



マイクロブログにおけるトピック出現量の時系列変化の抽出に関する研究

著者	福山 怜史
内容記述	筑波大学修士(情報学)学位論文・平成31年3月25日授与(41283号)
発行年	2019
URL	http://hdl.handle.net/2241/00159799

マイクロブログにおけるトピック出現量の時系列
変化の抽出に関する研究

筑波大学
図書館情報メディア研究科
2019年03月
福山 怜史

目次

第1章	はじめに	1
第2章	関連研究	4
2.1	潜在トピック推移の抽出に関する研究	4
2.2	マイクロブログからトピックを抽出することに関する研究	5
2.3	バースト現象に関する研究	5
第3章	Tweet Pooling による擬似文書を学習したトピックの出現量の時系列変化抽出	7
3.1	前準備	7
3.1.1	LDA における確率的変分推論	7
3.2	提案手法	8
3.2.1	擬似文書の作成	8
3.2.2	単位時間における推定ツイート数の計算	9
3.3	評価実験	9
3.3.1	実験データ	9
3.3.2	実験方法	10
3.3.3	実験結果	10
3.4	結論	12
第4章	Biterm topic model によって学習したトピックの出現量の時系列変化抽出	13
4.1	前準備	14
4.1.1	Biterm topic model	14
4.1.2	Biterm topic model における確率的周辺化変分推論 (SCVB0)	15
4.1.3	Biterm topic model におけるトピック分布のハイパーパラメータの最適化	17
4.2	提案手法	17
4.2.1	ミニバッチ学習によるトピック学習の高速化	17
	提案手法 1: BTM における SCVB0 によるミニバッチ学習	17
	提案手法 2: ミニバッチをさらに分割する学習 (ミニバッチ-セグメント学習)	18
4.2.2	単位時間あたりのトピック出現量の計算と近似	18
4.3	評価実験	20
4.3.1	実験データ	20
4.3.2	トピック学習の高速化の評価	21
	比較手法	21
	パラメータ設定	22
	実験方法	22
	実験結果	22

4.3.3	biterm の一部を抽出しトピック出現量を近似する方法の評価	24
	実験方法	24
	実験結果	25
4.4	結論	26
第 5 章	おわりに	28
	参考文献	30

目 次

3.1	ハッシュタグと Twitter で流行している話題の関係	11
4.1	文書から抽出される biterm の例	14
4.2	BTM のグラフィカルモデル	15
4.3	SCVB0 におけるミニバッチ学習とミニバッチ-セグメント学習. 提案手法 1 ではミニバッチ全体でパラメータ更新処理を行うことに対して, 提案手法 2 ではミニバッチをさらにセグメントに分割してセグメントごとにパラメータ 更新を行う.	19
4.4	ツイートに含まれる単語数ごとのツイート数	21
4.5	各手法の実行時間と平均対数尤度の変化	23
4.6	各手法の処理時間と RMSE の関係. グラフ上の点は表 4.2 の値をプロットし たものである.	25
4.7	各手法によって計算されるトピック出現量. 凡例は各トピックで出現確率の 高い語を順に左から並べたものである.	27

第1章 はじめに

Twitter は、現実世界で起こった事象に対してユーザがリアルタイムにツイートを投稿する性質から、現実世界を知覚するセンサとしての利用が期待されている。例えば、2011 年 3 月 11 日に発生した東日本大震災では、東京都において、地震発生から 1 時間以内に毎分 1,200 件以上のツイートが投稿されたことが報告されている [1]。また、近年では、Twitter のデータを利用した評判情報抽出 [2] や病気の流行予測 [3] といった手法の有効性が確認されており、Twitter ユーザが現実世界の事象に対して敏感に反応していることがわかる。このような背景から、特定の話題に関連するツイートの時間的な変化を抽出することによって、話題に対するユーザの興味関心をリアルタイムに分析できるようになることが期待されている。

本研究では、Twitter で観測される話題の出現量の時間的な変化について着目する。Twitter では、現実世界でユーザの関心を引きつけるような事象が発生した時に、その事象に関係する話題を含むツイートの投稿数が、局所的な時間で急激に増加する現象がしばしば観測される。このように、ツイートのような時間情報の与えられた文書集合においてその文書数が急激に増加する現象を“バースト現象”と呼ぶ [4]。バースト現象は、その話題に関係する事象に対する Twitter ユーザの関心のリアルタイムな変化が反映されていると考えられる。したがって、特定の事象に関係する話題を含むツイートの投稿数の時間的な変化を観察することによって、その事象に対する Twitter ユーザの関心の変化をリアルタイムに追跡することが可能となる。

特定の事象に関係する話題を含むツイートを取得するためには、あらかじめツイートの話題が何であるか特定する必要がある。しかし、Twitter では、あらかじめ定義されたタグ¹によってツイートが分類されていないため、それぞれのツイートに含まれる話題が何であるか知ることは容易ではない。加えて、Twitter におけるタグは、ハッシュタグと呼ばれる個々のユーザによってそれぞれのツイートにタグ付けが行われる分類手法²であるため、同義的なタグが大量に発生してしまう問題が報告されている [5]。したがって、ツイートに含まれる潜在的な話題から、その話題ごとにツイートをあらかじめ分類する必要があると考えられる。一般的に、文書に含まれる潜在的な話題は、ツイートの本文を構成する単語の組み合わせによって推定される。例えば、“オリンピック”に関係する話題では、行われる競技に関する“水泳”や“柔道”のような単語が出現しやすい傾向にある。したがって、話題に関する単語が含まれるツイートの投稿数をリアルタイムに追跡することによって、その話題の時間的な変化を捉えることができる。このような機能を提供するために、Twitter のスマートフォンアプリやウェブサイトでは、Twitter ユーザの間で流行しているキーワードがランキング形式で閲覧できるサービスが提供されている。このサービスを利用することによって、ユーザは Twitter で流行している話題について容易に調べることができる。しかし、流行している単語が出現する複数のツイートから、それらのツイートが示す話題を手作業で特定するためには多くの時間的なコストが掛かる。さらに、Twitter では、日々新しい話題が発生するため、それら全てを取り扱うことは困難である。このよう

¹分類項目や分類基準などを列挙して定義した分類手法であり、タクソノミーと呼ぶ

²フォクソノミーと呼ぶ

な背景から、本研究では、ツイートに含まれる潜在的な話題の抽出を自動的なアプローチによって行う。

文書に含まれる潜在的な話題の抽出を自動化する方法として、文書に含まれる潜在的な話題を確率的な生成モデルで表現し、この生成モデルの推論を行う手法が提案されている。この生成モデルを“トピックモデル”と呼ぶ。与えられた文書データに存在する潜在的な話題をモデル化し、そのモデルを推定する方法として、Bleiらは、Latent Dirichlet allocation (LDA) [6] と呼ばれる手法を提案している。この手法では、一つの文書に複数の潜在的なトピックが存在していることを仮定し、各トピックおよびトピックで発生する単語を多項分布でモデル化する。そしてこのモデルのパラメータを推論アルゴリズムによって推定し、それを文書データに潜在的に存在する話題の分布とそれぞれの話題の単語の分布とする。

また、時間情報の与えられた文書データに対して、LDAを拡張しトピックの時間発展を考慮した手法として、Dynamic topic model (DTM) [7] が提案されている。この手法では、LDAに対してトピックごとの時間経過による流行り廃れをコントロールするパラメータが与えられたモデルを仮定し、このトピックの流行り廃れの変化を推定することができる。この手法の応用例として、Koikeら [8] は、DTMを用いて、大量のニュース記事やツイートからトピックとそれらの時間的推移を抽出し、個々のトピックにおいてニュース記事やツイートがバーストしているか検出する手法を提案している。以上より、話題の抽出の自動化の観点では、LDAのようなデータから自動的にトピックを推定する手法が有効である。さらにLDAを拡張してトピックの時間発展を考慮したDTMでは、トピック出現量の時間的な変化を直接推定することができる。

一方で、LDAは、ツイートのような1文書に含まれる単語数が少ない文書データに対して学習が困難であることが指摘されている [9]。この問題は、LDAをベースにして構築された手法であるDTMでも同様に発生すると考えられるため、単語数が少ないデータに対しても有効な対策を適用する必要がある。またTwitterでは短時間に膨大な量のツイートが投稿されるため、モデルのパラメータを推定する学習アルゴリズムはデータサイズの影響を強く受けないものが望ましい。特に、特定の事象に関係する話題ごとのツイートの投稿数の時間的な変化を捉えるためには、数日分のツイートからトピックの学習を行う必要があるため、効率的な推論アルゴリズムが求められる。

本研究では、以上の課題を考慮し、1文書に含まれる単語数が少ないケースに特化し、効率的な推論アルゴリズムでトピックを学習し、抽出した話題の時間ごとのツイート投稿数を計算する手法を提案する。提案手法では、DTMのようにトピックとトピックの流行り廃れを同時に推定することはできないが、各ツイートに潜在的に含まれるトピックを推定し、あらかじめ決定した時間間隔ごとにそれぞれのトピックが割り当てられたツイートを集計することによって、それぞれの時間経過によるトピックの出現量の変化を得ることができる。本研究では、1文書に含まれる単語数が少ないケースに特化した手法として、Tweet Poolingによって複数のツイートを組み合わせた擬似文書をLDAに学習させる手法と、Biterm topic model (BTM) と呼ばれる、文書を非順序の二単語対 (biterm) に変換し、それぞれのbitermのトピックを推論する手法を利用する。提案手法では共通して、その時間に発生したツイートに対して抽出したトピックの割り当てを行い、トピックごとにそのツイート数を合計して単位時間あたりのトピック出現量として計算する。この時、1ツイートには複数のトピックが割合として推論されるため、単位時間に発生した全てのツイートからトピックごとにこの割合を合計した値が単位時間あたりのトピック出現量となる。またそれぞれの手法は、確率的最適化によるデータサイズの影響を強く受けない推論アルゴリズムを適用している [10] [11]。以上の手法により、ツイートに含まれる単語数が少ないことによって受ける悪影響を回避し、大量のツイートを効率的に処理してトピックの抽出を

行った上で、それぞれのトピックの時間経過による出現量の変化を計算することができる。これにより、既存の手法では困難であったツイートデータに含まれる潜在的なトピックの時間的な変化を容易に観察できるようになる。Tweet Pooling によって生成した擬似文書を LDA で学習させトピック出現量を計算する手法を 3 章に、BTM によって学習したトピックからその出現量を計算する手法を 4 章に論じる。

第2章 関連研究

本研究に関連して、時間経過による潜在トピックの変化を LDA を拡張することによってモデル化する研究、ツイートのような 1 文書に含まれる単語数が少ないケースにおけるトピック抽出に関する研究について紹介する。そして時間情報つき文書データにおけるバースト現象に関する研究を紹介し、ツイートとバースト現象の関連について説明する。

2.1 潜在トピック推移の抽出に関する研究

潜在トピック推移の抽出に関する研究として、Blei らによる Dynamic topic model (DTM) [7] が提案されている。DTM では、LDA に対して、トピックごとの単語分布およびトピック分布に時間マルコフ性を導入することによって、トピックの時間発展を捉えることを目的としている。時間発展させるパラメータは、トピックの流行り廃りを制御する α とトピックごとの単語分布を表す β であり、一時刻分のパラメータ遷移は正規分布を用いてモデル化されている。Blei らは、DTM によって、科学誌 Science に掲載された論文の時系列トピック解析を行い、異なる科学的テーマを含んだトピックおよびそのトピックで用いられる単語の傾向が確認できることを報告している。

Iwata らは、複数の時間スケールでのトピックの発展を解析する Multiscale dynamic topic model (MDTM) [12] を提案している。MDTM は、同一トピックにおいて出現する単語が時間スケールによって異なる性質を考慮したトピックモデルである。MDTM で考慮する時間スケール数を 1 つとした場合、MDTM は DTM と対応している。実験の結果、MDTM は時間スケール数を増加させることにより、モデルの平均パープレキシティが減少していることを確認し、複数の時間スケールの分布を考慮することの重要性を報告している。DTM や MDTM は、トピック分布のハイパーパラメータを時刻ごとに更新する手法であり、文書によらないグローバルな時間変化をモデリングしている。

また Iwata らは、LDA の各文書をそれぞれユーザの各文書と仮定し、文書ごとのトピック分布が独立して時間発展していることをモデル化した Topic tracking model (TTM) [13] を提案している。TTM と DTM および MDTM を比較すると、TTM はユーザごとの話題の関心の変化を捉えていることに対し、DTM や MDTM ではトピックの時間発展によるグローバルな変化を捉えている。以上より、同じ時間情報つき文書データに対しても、話題の時間発展がグローバルなものかあるいは文書ごとなものかを考慮して適切なモデルを選択する必要があるといえる。

一方でこれらの手法は、ベースとなる LDA と比較して、最適化するパラメータの数が多くなるため、学習により多くの時間が掛かってしまう可能性がある。また、LDA には Mimno らの提案する確率的変分推論 [10] による効率的な推論アルゴリズムが提案されている。この方法では、与えられた全ての文書データを一度に学習せずに一部の文書をミニバッチとしてサンプリングしながら効率的にトピックを学習することができる。Twitter のような短時間に大量のツイートが投稿されるケースの場合、一度に全てのツイートを推論せず、一部のツイートをサンプリングしながら学習する方が効率が良い。本研究では、ト

ピックの学習と時間経過によるトピックの変化の推定の処理を分離し、トピックの学習は既存手法である確率的変分推論による効率的なトピック学習の利用を行う。

2.2 マイクロブログからトピックを抽出することに関する研究

単語数の少ない文書のデータセットにおける問題として、Tang らは、非常に多数の文書があっても文書に含まれる単語が少なすぎると、LDA のパフォーマンスが低下する可能性があることを報告している [9]。この問題に対応するため、現在様々な手法が提案されている。Mehrotra ら [14] は、個々のツイートに含まれる単語が少数である問題を解決するために、Tweet Pooling を提案している。Tweet Pooling は、関連性のあるツイートを結合した擬似文書を作成し、これらの文書を LDA の学習用文書とする手法である。代表的な Tweet Pooling では、ユーザ・単位時間・バーストした語・ハッシュタグごとにツイートの結合が行われる。大量のツイートから話題の抽出を行う場合、擬似文書は特定の話題を示している必要があるため、特定の話題を示すタグであるハッシュタグごとにツイートを結合することが望ましい。Mehrotra らは、ハッシュタグごとにツイートを結合して擬似文書を作成する Hashtag Pooling が、クラスタリングおよびトピックの一貫性という観点において最も良い実験結果が得られたことを報告している。

ハッシュタグのクラスタリングに関連する手法として、Tsur ら [15] は、事象ごとにハッシュタグをクラスタリングする手法を提案している。この手法では、ハッシュタグごとにツイートを結合した擬似文書を作成し、その擬似文書をクラスタリングする手法を提案している。Tsur らは、特徴ベクトルの作成に TF-IDF ベクトルやハッシュタグの共起ベクトルを用いており、クラスタリング手法には k -means 法を用いている。このほか、井上ら [16] は、Tsur らが英語で行なったハッシュタグのクラスタリングが日本語でも適用できることを報告している。

Cheng ら [17] は、単語数の少ない文書データにおけるトピック学習のために、Biterm topic model (BTM) と呼ばれる LDA とは異なるモデルを提案している。BTM では、文書を biterm と呼ばれる非順序の二単語対の集合に変換し、各 biterm のトピックの推論を行う。BTM における biterm のトピック分布は、LDA のように文書ごとではなく、全ての biterm で共有されるため、文書ごとの単語数が短い場合でも単語の共起情報を十分に学習することができる。1 ツイートにおけるトピックの割合は、ツイートに含まれる biterm のトピックを集計することによって容易に計算できるため、BTM は LDA と同様に文書に複数のトピックが混合している仮定を保ちつつ文書に含まれる単語が少ない場合でも一貫性のあるトピックを抽出できる。また BTM は確率的な学習アルゴリズムが提案されており [11]、LDA と同様に全てのデータから一部をサンプリングして効率的にトピックを学習することができる。

2.3 バースト現象に関する研究

Twitter では、特定の話題が Twitter ユーザの間で流行した時、その話題に関係するツイートの投稿が短時間の間に急激に増加する。この現象はブログやニュース記事のような時間情報がついた文書でも同様に発生し、その原因は特定の事象が発生したことをきっかけにその事象に関係する文書の投稿が爆発的に増加するためである。このような時間情報の与えられた文書データの投稿数の増加を“バースト現象”と定義し、この現象を検出する手法の一つとして、Kleinberg はバースト解析アルゴリズム [4] を提案している。この手法

は、時系列データに対して定常状態とバースト状態を確率的オートマトンによってモデル化し、それぞれの状態において異なるパラメータを持つポアソン分布を仮定し、記事の出現頻度の増加・減少にしたがって、それぞれの状態を行き来することによって、現在の時系列データの状態を決定する。この手法は、システムのログデータなどの普段の文書の発生頻度が一定な場合において、突如投稿頻度が急上昇する状態を検出することに対して有効な手法である。しかし、特定の話題を含む記事では投稿頻度が緩やかに増減する場合があるため、それぞれの状態を定常状態と仮定する Kleinberg のバースト解析アルゴリズムでは必ずしも正確にバースト検出できない可能性がある。

次に、Twitter におけるバースト現象の分析を行った研究として、水沼ら [18] の研究がある。水沼らは、Twitter におけるバーストの特徴を分析し、その特徴ごとにバーストの類型化を行なっている。水沼らはツイートのバースト検出手法を選択するために、バースト現象を出現頻度の外れ値とみなし、外れ値検知手法である ROKU [19]・ 3σ 法・増山の検定 [20]・MAD 法 [21] [22] の比較を行なっている。この比較では、1 度のバースト現象を複数のイベントとみなすことがないか、4 手法で共通して得られる確信度の高いバースト現象がどれぐらい多く検出できるかを評価している。この結果、 3σ 法が、バースト平均継続時間が最も長く、また 4 手法がいずれもバースト現象として検出したハッシュタグの集合との一致率が最も高いことから、4 手法の中で最も理想的な手法であるとしている。

また単語数の少ない文書データにおいて有効なトピックモデルの手法である BTM に対して、バーストしているトピックと日常的に発生するトピックを同時に学習するモデルとして Bursty BTM (BBTM) [23] が提案されている。この手法では、burstiness と呼ばれるバースト現象の強さを BTM のモデルに導入することで、バーストしやすいトピックと日常的に発生するトピックを分離する。この手法ではトピック出現量が緩やかに上昇するトピックには対応していないため、バーストしているトピックのみ抽出することを目的としている分析において有用であると考えられる。

第3章 Tweet Poolingによる擬似文書を学習したトピックの出現量の時系列変化抽出

Tweet Pooling は、ツイートデータを LDA で学習させる際に、関係性のあるツイートごとに個々のツイートを結合させた擬似文書を LDA の学習用データとする手法である。Mehrotra ら [14] は Tweet Pooling の各手法によって獲得した擬似文書を学習させたトピックにおいて、Hashtag Pooling がトピックのまとまりが最も良いことを確認した。提案手法では、Hashtag Pooling によって生成した擬似文書から LDA でトピックを抽出し、抽出したトピックの単位時間あたりの出現量を推定する。この時、Hashtag Pooling による擬似文書のトピックの推論を行なった場合、1 ハッシュタグにおけるトピックの割合しか計算できない。そこで本研究では、新たな Tweet Pooling の手法として、Hashtag Pooling による擬似文書を単位時間ごとに分割する手法を提案する。これにより、生成された擬似文書は単位時間あたりに発生したハッシュタグのツイートの集合とみなすことができるため、この擬似文書のトピック割合と擬似文書を生成するために結合したツイートの数の積を、単位時間あたりの 1 ハッシュタグにおけるトピック出現量として計算することできる。そして、単位時間あたりの 1 ハッシュタグにおけるトピック出現量を全てのハッシュタグで同様に計算することによって、単位時間におけるトピック出現量が計算される。

本章では、はじめに LDA の効率的な推論アルゴリズムについて紹介し、次に提案手法について説明する。そして提案手法に対して実際に投稿されたツイートをを用いた実験とその結果について述べる。本研究では、実際のツイートを対象に、提案手法によってトピック出現量を計算し、そのトピック出現量の時間的な変化が Twitter ユーザのその話題への興味関心を反映しているか確認を行う。実験では、提案手法によって計算したトピック出現量から、バースト検出手法によってバーストしている時刻を検出し、トピックに関連する現実世界の事象がバースト時刻に近い時刻で発生しているか調査する。バースト検出手法は、データの分布を仮定しない異常検知の手法である 3σ 法を用いる。

3.1 前準備

3.1.1 LDA における確率の変分推論

提案手法では、LDA の推論アルゴリズムとして Mimno らの提案する確率の変分推論 [10] を用いる。この手法では確率的最適化を行なっているため、全てのデータを使わず、一部のサンプルを使って逐次的にトピックを学習することができる。ツイートデータはサンプル数が膨大であり、既存の学習アルゴリズム [6] [24] では計算コストが高いことが予想されるため、一部のサンプルから逐次的に学習を行う手法が効率的であるといえる。Mimno らの手法では、トピック k の単語分布のハイパーパラメータ λ_k について単語 w におけるパラメータを式 (3.1) の反復計算によって最適解を得る。

$$\lambda_{k,w}^{(s)} = (1 - \nu^{(s)})\lambda_{k,w}^{(s-1)} + \nu^{(s)} \left(\beta_{k,w} + \frac{D}{B} \sum_{d \in D^{(s)}} \mathbb{E}_q[n_{d,k,w}] \right). \quad (3.1)$$

更新式における D は学習の対象となる文書 X の数である．また B は一度の学習に使用するサンプルの数でありミニバッチサイズと呼ぶ．この更新式は，確率的最適化の一種である Robbins–Monro 法を適用しており， $\nu^{(s)}$ はこの更新式が収束を保証する条件 $\sum_{s=1}^{\infty} \nu^{(s)} = \infty$ および $\sum_{s=1}^{\infty} (\nu^{(s)})^2 < \infty$ を満たすような数列である．本研究ではこの $\nu^{(s)}$ を式 (3.2) で計算する [25]．

$$\nu^{(s)} = \frac{1}{(\tau + s)^\kappa}, \quad (3.2)$$

$$\begin{aligned} \text{where} \quad & \tau > 0, \\ & 0.5 < \kappa \leq 1. \end{aligned}$$

この時， $\mathbb{E}_q[n_{d,k,w}]$ は文書 X_d においてトピック k が単語 w に割り当てられる回数の期待値であり，文書 X_d のトピック分布に基づいて計算される．本研究では，Nozawa ら [26] が行なった手法に基づき文書 X_d のトピック分布を Gibbs sampling によるトピック $z_{d,i}$ の分布 $q(z_{d,i} = k | z_d^{\setminus i})$ から，近似的に同時確率を求める．Nozawa らは，Wang ら [27] の提案する Locally Collapsed 近似法を用いた式 (3.3) からトピック $z_{d,i}$ のサンプリングおよび分布 $q(z_{d,i} = k | z_d^{\setminus i})$ の更新を行い， M 回の更新後に得られる文書 X_d に出現する単語 w のトピックを用いて同時確率を近似している．

$$q(z_{d,i} = k | z_d^{\setminus i}) \propto (\alpha_k + n_{d,k}) \frac{\lambda_{k,x_{d,i}}}{\sum_v \lambda_{k,w}}. \quad (3.3)$$

このサンプリング中に，トピック k が単語 w に割り当てられている数を $\mathbb{E}[n_{d,k,w}]$ としてサンプル近似する．

3.2 提案手法

3.2.1 擬似文書の作成

提案手法による擬似文書は，Hashtag Pooling による擬似文書を単位時間ごとに分割したものである．したがって，ツイートに対しては，ハッシュタグかつ単位時間ごとにツイートを結合し，擬似文書を生成する．時刻 t におけるハッシュタグの集合を H_t とする．ここでは， H_t による擬似文書を LDA の学習データとする．

トピックモデルの学習データに用いる擬似文書の表現方法として，共起する単語の Bag of Words を用いる．期間 T における時刻を t とする．ある時刻 t におけるハッシュタグ $h_t \in H_t$ が出現するツイートの集合を D_{h_t} とする．ツイート $d_{h_t} \in D_{h_t}$ に語彙 w_i が出現する頻度を $tf(w_i, d_{h_t})$ と表すと，時刻 t におけるハッシュタグ h_t と共起する単語の頻度は以下のように定義される．

$$tf(w_i, h_t) = \sum_{d_{h_t} \in D_{h_t}} tf(w_i, d_{h_t}). \quad (3.4)$$

以上の方法で全てのハッシュタグにおける語の出現頻度を LDA の学習データとする．

3.2.2 単位時間における推定ツイート数の計算

擬似文書に対応する LDA によって推論されたトピック $k \in K$ の分布を θ_{h_t} とする． h_t におけるトピック k の推定ツイート数 N_{k,h_t} は式 (3.5) によって計算される．

$$N_{h_t,k} = N_{h_t} \cdot \theta_{h_t,k}. \quad (3.5)$$

計算された推定ツイート数 N_{k,h_t} を基に，時刻 t におけるトピック k の推定ツイート数 $N_{t,k}$ を式 (3.6) によって計算する．

$$N_{t,k} = \sum_{h_t \in H_t} N_{h_t,k}. \quad (3.6)$$

この計算の結果得られたトピック k の時系列データ $N_k = (N_{1,k}, N_{2,k}, \dots, N_{t,k}, \dots, N_{|T|,k})$ を期間 T におけるトピック k の推定ツイート数とする．

3.3 評価実験

本研究では，提案手法によって推定したトピック出現量の有効性を確かめるため，推定したトピック出現量に対してバースト現象の検出を行う．バースト現象は，時系列データにおいて局所的な時間にその値が急激に増加する現象であり，Twitter では特定の事象が発生したときにその事象に関連する話題を含むツイートの投稿数が急激に増加する現象として観測される．本研究では，提案手法によって推定したトピック出現量に対してバースト現象の検出を行い，バースト現象の発生時刻とそのトピックが関連する事象の発生時刻に相関関係があるかどうか確認を行う．この実験によってバースト現象の発生時刻とそのトピックが関連する事象の発生時刻に相関関係が見つかった場合，提案手法によって推定されたトピック出現量は Twitter ユーザの話題に対する関心を反映していると考えられる．以上の観点より，実際のツイートデータに対して提案手法で得られたバーストしているトピックが，当該期間にて発生した現実世界の事象と共起しているか確認を行った．

3.3.1 実験データ

検証では，単語分布の次元数が膨大になることを防ぎ，かつ，特定の話題を表す傾向の強い品詞に限定することを目的として，LDA の学習データである擬似文書および擬似文書に出現する語彙に以下の制約を設ける．

1. 語彙の品詞は固有名詞・普通名詞・サ変接続の名詞のみ．
2. 語彙の文字列長は・漢字は 1 字以上・ひらがな・カタカナ・数字・記号では 2 字以上．
3. リツイートを示す「RT」・URL・リプライを示す「@ユーザ名」の文字列は無視．
4. 任意の時刻において (1) から (3) の条件を満たす語彙を含むツイート数が 10 以上のハッシュタグ．ただし 10 未満の場合は当該時刻においてそのハッシュタグは考慮しない．

これらの条件を満たすハッシュタグは，2012 年 8 月 1 日から 2012 年 8 月 7 日の間に存在したハッシュタグ 965,369 種類中 40,851 種類，各ハッシュタグのコーパスに用いるツイートはハッシュタグと同一の期間に発生した 26,639,495 ツイート中 17,294,548 ツイート，擬似文書数は 117,547 であった．

表 3.1: 提案手法によって得られるトピックの例

トピック	α	単語 $\phi_{k,w}$							
1	0.009	体操	0.319	内村	0.208	演技	0.043	競技	0.031
2	0.008	北島	0.127	競泳	0.116	介	0.079	康	0.071
3	0.007	広島	0.472	野田	0.087	黙禱	0.041	投下	0.031
4	0.004	エジプト	0.128	ブラジル	0.126	永井	0.094	吉田	0.067
5	0.003	浜田	0.149	死去	0.141	幸一	0.135	衆議院	0.087

表 3.2: トピックの推定ツイート数の推移とバースト時刻

トピック	1 日	2 日	3 日	4 日	5 日	6 日	7 日	バースト時刻 (日)
1	4,412	24,011	3,793	1,981	4,950	2,843	2,244	8 月 2 日
2	4,685	11,708	5,156	1,351	9,403	1,757	2,463	8 月 2 日
3	830	849	1,247	1,134	1,370	6,602	1,603	8 月 6 日
4	1,117	5,333	1,141	37,554	5,506	1,058	4,061	8 月 4 日
5	272	841	599	332	10,954	1,119	541	8 月 5 日

実験環境は、OS が Ubuntu 16.04, CPU が Intel Xeon E5-2630 (2.40GHz) 8core 2 機である。また、実装は Python および Java で行う。形態素解析器は MeCab [28] を使用し、LDA は Mimno ら [10] の確率的変分推論のアルゴリズムを Java によって実装する。

3.3.2 実験方法

3.2.2 節の方法によって得られた時系列データ $N_k = (N_{1,k}, N_{2,k}, \dots, N_{t,k}, \dots, N_{|T|,k})$ に対して、以下の条件式を満たす時刻 t をバースト時刻とする。

$$N_{t,k} > E[N_{k \setminus t}] + 3\sqrt{V[N_{k \setminus t}]} \quad (3.7)$$

実験では、時系列の単位を 1 日とし、当該時刻 t 以外のデータから平均と標準偏差を求め、 3σ 法による閾値を計算する。ここで時系列の単位は、ツイート全体の投稿数が増加しやすい時間帯では誤ってバースト検出される可能性を考慮し現状の制約として 1 日と設定する。次に 3σ 法によるバースト検出では、過去の出現頻度が 0 の場合、推定ツイート数が 1 ツイートでも観測されればバースト検出されてしまう問題がある。例えば、 $N_k = (0, 0, 0, 0, 0)$ の時系列データが与えられた場合、推定ツイート数が 1 ツイートであってもバースト現象と検出される。この問題を考慮し、本研究では、トピックの推定ツイート数が少なくとも 100 を超える場合のみ、バースト検出することとする。

LDA の各パラメータの設定については、トピック数が 10,000, イテレーション回数が 1,000 回, ミニバッチサイズが 1,000 とする。この結果、トピック学習に掛かる処理時間は 143 分であった。

3.3.3 実験結果

実験データから提案手法によって得られたトピックの一部を表 3.1 に示す。表 3.1 に示すトピックは当該期間においてバースト現象が検出されたトピックであり、同様にバースト現象が検出されたトピックは 10,000 トピック中 647 トピックであった。 α は各文書におけ

表 3.3: バースト時刻における各ハッシュタグの推定ツイート数

トピック	ハッシュタグ 推定ツイート数							
1	体操	5,440	オリンピック	4,258	olympic	975	mitazo	964
2	オリンピック	3,888	競泳	3,430	olympic	502	水泳	308
3	広島	472	原爆	374	twitr	374	IWJ_HIROSHIMA2	266
4	daihyo	6,880	オリンピック	5,522	サッカー	4,560	なでしこ	2,426
5	nhk_news	4,771	2ch	627	news	615	niconews	467

るトピック分布の事前分布であるディリクレ分布のハイパーパラメータであり、全体と比較した際のトピックの出現のしやすさとみなすことができる。また $\phi_{k,w}$ は、トピック k における語 w の出現する確率であり、この確率の高い上位 4 語を表 3.1 に示す。

表 3.4: トピックと対応する現実世界の事象

トピック	事象	発生時刻 (日)
1	ロンドン五輪男子体操個人総合	8月2日
2	ロンドン五輪男子平泳ぎ北島康介氏 4 位入賞	8月2日
3	広島原爆の日 平和記念式典に野田佳彦氏参列	8月6日
4	ロンドン五輪男子サッカー準々決勝 日本対エジプト戦	8月4日
5	浜田幸一氏逝去	8月5日

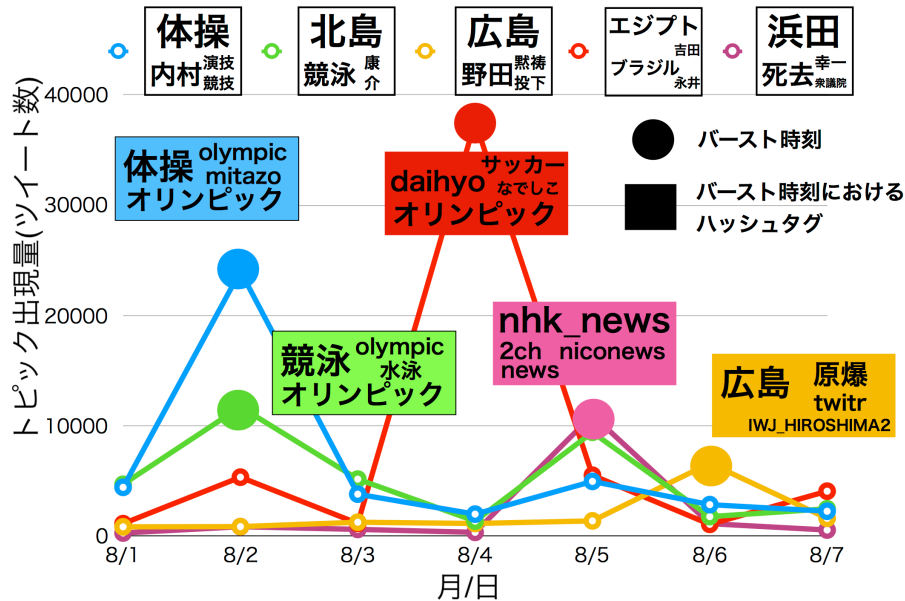


図 3.1: ハッシュタグと Twitter で流行している話題の関係

次に、表 3.1 のトピックの推定ツイート数の推移とバースト時刻を表 3.2 に示す。バースト時刻においては、いずれのトピックも突出した推定ツイート数となっており、この時刻においてトピックでバースト現象が発生していることがわかる。

この結果をもとに、バースト時刻において当該トピックに関する推定ツイート数の多い上位 4 つのハッシュタグを表 3.3 に示す。トピック 1・2・4 では、共通してオリンピックに

関連するハッシュタグである“オリンピック”や“olympic”が割り当てられている。またトピック3では、“広島”や“原爆”など単語としても共起しやすい語がハッシュタグが割り当てられている。トピック5では、 $\phi_{k,w}$ の高い語とは異なり、トピック5のカテゴリーを示すニュースに関連するハッシュタグが割り当てられている。以上の結果から、トピックに対応する単語とハッシュタグは同様の傾向がある語やその話題のカテゴリーが割り当てられやすいことがわかる。この性質を利用することによって、単語とハッシュタグによる多面的なトピックの解釈が可能となると考えられる。

実験結果から、トピックと関連性があると考えられる現実世界の事象を表3.4に示す。このように提案手法では、バーストしているトピックの情報として、トピック内で出現しやすい単語、バーストした時刻、トピックが割り当てられたハッシュタグといった多様な情報を、図3.1に示すことによって、事象を特定することができる。

3.4 結論

本章では、LDAのアルゴリズムを変更せずに、Tweet Poolingの拡張によって、潜在的なトピックの推移を抽出する手法を提案した。実際に提案手法を用いて、2012年8月1日から8月7日に発生した現実世界の事象に対応するトピックの抽出とそれらのトピックの推移の抽出とバースト検出を行った。この結果、LDAの高速な推論アルゴリズムが利用できるとともに、トピックに出現しやすい語・バースト時刻・トピックが割り当てられたハッシュタグといったトピックに関する豊富な情報から、容易に現実世界の事象との関係性を分析できることがわかった。

第4章 Biterm topic modelによって学習したトピックの出現量の時系列変化抽出

Biterm topic model (BTM) [17] は、文書を biterm と呼ばれる非順序の二単語対に変換し、各 biterm のトピックを推論することによって文書に含まれるトピックを学習する手法である。BTM ではグローバルな潜在変数によってトピックが学習されるため、短文書データセットにおいて、LDA と比較して一貫性の高いトピックを抽出できることが報告されている。提案手法では、BTM によってツイートデータからトピックを抽出し、抽出したトピックの単位時間あたりの出現量を推定する。抽出したトピックの単位時間あたりの出現量は、1 ツイートあたりのトピック割合から単位時間に出現した全てのツイートで集計を行うことによって計算される。BTM は、Hashtag Pooling [14] のようにハッシュタグ付きツイートに限定せず、全てのツイートを取り扱えることから、Twitter を利用する全ユーザーが関心を持つ話題を網羅的に抽出することができると考えられる。

一方で、ツイートに含まれる biterm を網羅的に抽出した場合、トピック学習時に膨大な数の biterm を取り扱う必要があるため、効率的な推論アルゴリズムを適用する必要がある。この問題を考慮して、本研究では確率的周辺化変分推論 (Stochastic collapsed variational Bayesian inference; SCVB0) [11] と呼ばれる推論アルゴリズムによって、biterm 全体から一部の biterm をサンプリングしながら学習する手法を検討する。しかし、これまでに提案されている SCVB0 のアルゴリズムは biterm を一つずつサンプリングする手法であるため、一貫性の高いトピックを抽出するために膨大な回数の反復処理が必要である。また、BTM によって抽出したトピックの単位時間あたりの出現量は、各時間内に投稿されたツイートに含まれる biterm のトピックを推論することによって計算されるが、投稿されたツイートに含まれる全ての biterm に対する推論処理は時間的なコストが高い。

本研究では、SCVB0 に対して、ミニバッチ学習をベースとした拡張と近似的なトピック出現量の計算によって、より高速な学習とトピック出現量の計算を行う手法を提案する。提案手法では、BTM を高速化する手法として、SCVB0 におけるミニバッチ学習を提案する。ミニバッチ学習とは、学習するデータの中から複数のデータをサンプリングし、それを全体のデータの近似として学習する手法である。本論文ではミニバッチ学習を適用することにより、効率的な学習アルゴリズムが導出できることを示す、また近似的なトピック出現量の計算手法として、単位時間において投稿されたツイートに含まれる biterm の内、一部だけトピック出現量の計算に使用する手法を提案する。この手法によって、トピック出現量には誤差が含まれてしまうが、単位時間に出現する全ての biterm からトピック出現量を計算する場合と比較してより高速に計算することができる。

本章では、はじめに提案手法のベースとなる BTM とその効率的な推論アルゴリズムについて紹介し、次に提案手法について説明する。そして提案手法に対して実際に投稿されたツイートをを用いた実験とその結果について述べる。実験は、トピックの学習およびトピック出現量の計算の高速化の二つの取り組みについてそれぞれ行なう。

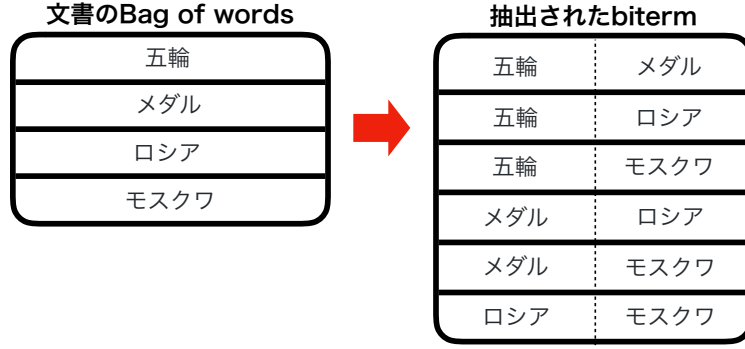


図 4.1: 文書から抽出される biterm の例

4.1 前準備

4.1.1 Biterm topic model

Biterm topic model (BTM) [17] とは、トピックモデルの一種であり、文書を biterm と呼ばれる非順序の二単語対の集合に変換し、各 biterm のトピックを推論することによってトピックを学習する手法である。この手法は、文書に含まれる単語数が少数のデータセットに対して、質の高いトピックを抽出することに特化している。Cheng らは、BTM によって推定されるトピックが、既存のトピックモデルと比較して、トピックに出現する語の一貫性がより高いことを報告している。

本研究では、Cheng らの方法 [17] を参考に、次の手順によって文書から biterm を生成する。はじめに文書を Bag of Words (BoW) に変換し、この BoW から単語の組み合わせを抽出する。この時、文書内に同一の単語が複数存在することによって出現する同じ単語の組み合わせは除外する。実際の文書から biterm を抽出する例を図 4.1 に示す。

BTM で用いられる変数の表記法を以下に示す。

- $N_{\mathbf{B}}$ 個の biterm の集合を $\mathbf{B} = \{b_i\}_{i=1}^{N_{\mathbf{B}}}$ とする。ただし $b_i = \{w_{i,1}, w_{i,2}\}$ 。
- biterm b_i のトピックを z_i とする。
- K 次元のトピック分布を示すベクトルを $\Theta = \{\theta_k\}_{k=1}^K$ とする。ただし、ベクトルの L_1 ノルムの大きさは 1 とする。
- サイズ $K \times W$ の単語分布の行列を $\Phi = \{\phi_k\}_{k=1}^K$ とする。行列の各行ベクトルはトピックごとの単語分布を表し、その次元は W かつ L_1 ノルムの大きさは 1 とする。

BTM では以下の過程に従って biterm が生成されることをモデル化している。

1. Draw $\theta \sim \text{Dirichlet}(\alpha)$.
2. For each topic $k \in [1, K]$,
 - (a) Draw $\phi_k \sim \text{Dirichlet}(\beta)$.
3. For each biterm $b_i \in \mathbf{B}$,
 - (a) Draw $z_i \sim \text{Multinomial}(\theta)$.
 - (b) Draw $w_{i,1}, w_{i,2} \sim \text{Multinomial}(\phi_{z_i})$.

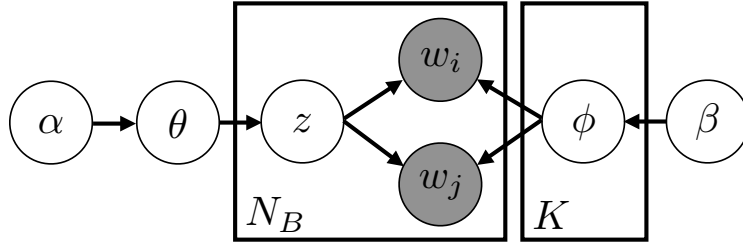


図 4.2: BTM のグラフィカルモデル

この生成過程のグラフィカルモデルを図 4.2 に示す. BTM を考案した Cheng らはこのモデルを周辺化ギブスサンプリング (Collapsed Gibbs Sampling; CGS) によって推論を行う手法を提案している [17].

4.1.2 Biterm topic model における確率的周辺化変分推論 (SCVB0)

BTM を効率的に推論する確率的な推論アルゴリズムとして確率的周辺化変分推論 (Stochastic collapsed variational Bayesian inference; SCVB0) が提案されている [11]. この手法は, LDA における周辺化変分推論 (Collapsed variational Bayesian inference; CVB) [29] に対してトピックの期待値の式のテイラー展開を 0 次に近似したアルゴリズム (CVB0) [30] を, BTM の推論アルゴリズムに適用し, 確率的な推論アルゴリズムに拡張したものである. Awaya らは, BTM における SCVB0 の推論アルゴリズムが Cheng らの提案する周辺化ギブスサンプリング [17] よりトピックの学習が高速であることを報告している [11]. BTM における SCVB0 では, 任意の biterm b のトピック z_i が k である確率を示す変分パラメータ $\hat{z}_{i,k}$ を式 (4.1) で推定する.

$$\hat{z}_{i,k} = P(z_i = k | z, \mathbf{B}) \propto (N_k + \alpha) \frac{(N_{w_{i,1}|k} + \beta)(N_{w_{i,2}|k} + \beta)}{(2N_k + W\beta)(2N_k + W\beta + 1)}. \quad (4.1)$$

(4.1) で biterm b_i のトピックが計算されるたびに, コーパスにおけるトピック k の出現頻度の期待値 N_k とトピック k で単語 w が出現する頻度の期待値 $N_{w|k}$ の推定値として \hat{N}_k および $\hat{N}_{w|k}$ をそれぞれ式 (4.2) および (4.3) によって計算する.

$$\hat{N}_k = |\mathbf{B}| \hat{z}_{i,k}, \quad (4.2)$$

$$\hat{N}_{w|k} = \begin{cases} |\mathbf{B}| \hat{z}_{i,k} & \text{if } w \in b_i, \\ 0 & \text{otherwise.} \end{cases} \quad (4.3)$$

確率的最適化に則り, この期待値の近似値から N_k と $N_{w|k}$ を式 (4.4) および (4.5) によって計算する.

$$N_k \leftarrow (1 - \nu^{(s)})N_k + \nu^{(s)}\hat{N}_k, \quad (4.4)$$

$$N_{w|k} \leftarrow (1 - \nu^{(s)})N_{w|k} + \nu^{(s)}\hat{N}_{w|k}. \quad (4.5)$$

$\nu^{(s)}$ は, LDA における確率的変分推論 [10] の場合と同様に, 式 (3.2) で計算する. BTM における SCVB0 による推論アルゴリズムを Algorithm 1 に示す.

Algorithm 1 BTM における SCVB0 による推論アルゴリズム

Randomly initialize N_k and $N_{w|k}$
for $b_i \in \mathbf{B}$ **do**
 for each topic k **do**
 Compute $\hat{z}_{i,k}$ using Eq. (4.1)
 Update N_k using Eq. (4.4) and $N_{w|k}$ using Eq. (4.5)
 end for
end for
Compute global parameters Θ using Eq. (4.6) and Φ using Eq. (4.7)

Algorithm 1 における計算複雑性は $\mathcal{O}(W)$ である。Awaya らは、 $N_{w|k}$ の計算複雑性を $\mathcal{O}(1)$ にする手法を提案している。この手法では、 s 回目の行列の更新における $N_{w|k}$ をスカラーと行列の積の形である $\rho_s \tilde{N}_{w|k}$ とする。 ρ_s は $\prod_{i=1}^s (1 - \nu^{(s)})$ によって計算される。これは式 (4.5) の計算で全ての単語に対して共通して乗算される定数に対して s 回イテレーションを行なった後の積である。この定数を用いることによって、biterm b に含まれる単語 w_1, w_2 の更新は $\tilde{N}_{w|k}$ の要素 (k, w_1) および (k, w_2) に対する $\frac{\nu^{(s)}|\mathbf{B}|}{\rho_s}$ の加算のみとなる。ただし、イテレーション回数 s が大きくなった場合、 ρ_s の値が非常に小さくなり数値的に不安定になるため、 ρ_s の値が閾値を下回った時点で配列内の全ての要素に ρ_s を積算する。この時、 ρ_s およびイテレーション回数 s は、リセットされる。本研究では、SCVB0 におけるトピック割合 Θ とトピックの単語分布 Φ を N_k と $N_{w|k}$ を用いてそれぞれ式 (4.6) および (4.7) によって計算する。 N は $\sum_{k'=1}^K N_{k'}$ 、 $N_{\cdot|k}$ は $\sum_{w'=1}^W N_{w'|k}$ の計算を表す。

$$\theta_k = \frac{N_k + \alpha}{N + \sum_{k'=1}^K \alpha_{k'}}, \quad (4.6)$$

$$\phi_{k,w} = \frac{N_{w|k} + \beta}{N_{\cdot|k} + W\beta}. \quad (4.7)$$

文書ごとのトピック割合は、文書に含まれる biterm のトピックをもとに計算される。 N_d 個の biterm を含む文書 d を $\{b_i^{(d)}\}_{i=1}^{N_d}$ とした時、文書 d のトピック割合は式 (4.8) によって計算される [17].

$$P(z|d) = \sum_{i=1}^{N_d} P(z|b_i^{(d)})P(b_i^{(d)}|d). \quad (4.8)$$

biterm $b_i^{(d)} = (w_{i,1}^{(d)}, w_{i,2}^{(d)})$ が与えられた時、この $b_i^{(d)}$ のトピック割合 $P(z|b_i^{(d)})$ は式 (4.9) によって計算される。

$$P(z = k|b_i^{(d)}) = \frac{\theta_k \phi_{k,w_{i,1}^{(d)}} \phi_{k,w_{i,2}^{(d)}}}{\sum_{k'} \theta_{k'} \phi_{k',w_{i,1}^{(d)}} \phi_{k',w_{i,2}^{(d)}}}. \quad (4.9)$$

また、文書中の biterm の割合 $P(b_i^{(d)}|d)$ は式 (4.10) によって推定される。

$$P(b_i^{(d)}|d) = \frac{n(b_i^{(d)})}{\sum_{i=1}^{N_d} n(b_i^{(d)})}. \quad (4.10)$$

$n(b_i^{(d)})$ は文書 d 中の biterm $b_i^{(d)}$ の頻度である。

Algorithm 2 BTM における SCVB0 による推論アルゴリズム (ミニバッチ学習)

```
Randomly initialize  $N_k$  and  $N_{w|k}$ 
for iterations do
  Sample biterms minibatch  $\mathbf{B}^{(s)}$ 
  Compute  $N_b^{(s)}$  in  $\mathbf{B}^{(s)}$ 
  for  $b_v \in \mathbf{B}_{\text{unique}}^{(s)}$  do
    for each topic  $k$  do
      Compute  $\hat{z}_{v,k}$  using Eq. (4.1)
    end for
  end for
  Update  $N_k$  using Eq. (4.12) and  $N_{w|k}$  using Eq. (4.13)
end for
Compute global parameters  $\Theta$  using Eq. (4.6) and  $\Phi$  using Eq. (4.7)
```

4.1.3 Biterm topic model におけるトピック分布のハイパーパラメータの最適化

BTM において、トピック分布 Θ およびトピック k の単語分布 ϕ_k はそれぞれ $\alpha \cdot \beta$ をハイパーパラメータとしたディリクレ分布を事前分布に持つ。Wallach ら [31] は、LDA においてトピック分布 Θ のハイパーパラメータ α は各要素が異なる値を持つ非対称 Dirichlet 分布、トピック k の単語分布 ϕ_k のハイパーパラメータ β は各要素が同一の値を持つ対称 Dirichlet 分布が有用であることを報告している。

これを考慮し、本研究ではトピック分布 Θ のハイパーパラメータ α の最適化を行う。本研究では、BTM の学習と同時にハイパーパラメータ α の最適化を行い、後述する提案手法では、不動点反復法による式 (4.11) の反復計算によって更新を行う。

$$\alpha_k \leftarrow \alpha_k \frac{\Psi(N_k + \alpha_k) - \Psi(\alpha_k)}{\Psi(N. + \sum_{k'=1}^K \alpha_{k'}) - \Psi(\sum_{k'=1}^K \alpha_{k'})}. \quad (4.11)$$

4.2 提案手法

提案手法では、トピックの学習の高速化のために、SCVB0 をミニバッチ学習に拡張する。また、抽出したトピックの出現量を推定する時、単位時間に出現した biterm から一部に対して推論を適用することで高速化を図る。

4.2.1 ミニバッチ学習によるトピック学習の高速化

提案手法 1 : BTM における SCVB0 によるミニバッチ学習

訓練するデータの中から複数のデータをサンプリングし、それを全体のデータの近似として学習する手法をミニバッチ学習と呼ぶ。提案手法 1 では、BTM を SCVB0 のアルゴリズムで学習する際に biterm 全体からあらかじめ決定したミニバッチサイズの数だけ biterm をサンプリングし、ミニバッチごとに N_k と $N_{w|k}$ の更新を行う。

提案手法 1 では、ミニバッチ内で biterm が重複して含まれる性質を考慮し、より効率的な手法を提案する。SCVB0 によるミニバッチ学習のアルゴリズムを Algorithm 2 に示す。Algorithm 2 における SCVB0 の更新式を式 (4.12) および (4.13) とする。

$$N_k \leftarrow (1 - \nu^{(s)})N_k + \nu^{(s)} \frac{|\mathbf{B}|}{|\mathbf{B}^{(s)}|} \sum_{b_v \in \mathbf{B}_{\text{unique}}^{(s)}} \hat{z}_{v,k} N_{b_v}^{(s)}, \quad (4.12)$$

$$N_{w|k} \leftarrow (1 - \nu^{(s)})N_{w|k} + \nu^{(s)} \frac{|\mathbf{B}|}{|\mathbf{B}^{(s)}|} \sum_{b_v \in \mathbf{B}_{\text{unique}}^{(s)}} \hat{z}_{v,k} N_{b_v}^{(s)} \delta(w \in b_v). \quad (4.13)$$

$\mathbf{B}^{(s)}$ は s 回目のイテレーションにおけるミニバッチに含まれる biterm の集合、 $\mathbf{B}_{\text{unique}}^{(s)}$ は V 個のユニークな biterm の集合、 $\mathbf{B}_{\text{unique}}^{(s)}$ は s 回目のイテレーションにおけるミニバッチに含まれるユニークな biterm の集合、 $N_{b_v}^{(s)}$ はミニバッチに含まれるユニークな biterm b_v の頻度を表す。Algorithm 2 においてミニバッチとして複数の biterm をサンプリングした際に、ミニバッチ内に出現する各 biterm の頻度を数え上げ、ミニバッチ内のユニークな biterm b_v の変分パラメータ $\hat{z}_{v,k}$ とその biterm の頻度 $N_{b_v}^{(s)}$ の積の総和を N_k と $N_{w|k}$ の更新量として計算する。これにより、学習のボトルネックとなるミニバッチ内の各 biterm の変分パラメータの計算回数は、ミニバッチ内に含まれる異なり biterm 数まで削減される。

提案手法 2：ミニバッチをさらに分割する学習 (ミニバッチ-セグメント学習)

本研究では、ミニバッチ内に重複する biterm が存在する性質を利用し、さらに効率的な学習手法を提案する。4.2.1 節で説明しているミニバッチ学習では、ミニバッチサイズが大きければ大きいほど重複する異なり biterm の種類の数が多くなるため、変分パラメータの計算回数はより削減される。一方で、ミニバッチサイズを単純に大きくすればするほど、1 つのミニバッチを学習するために必要な処理時間は増大すると考えられる。

提案手法 2 では、ミニバッチサイズを大きく保ちながら処理時間を短縮させるために、ミニバッチ内の biterm の頻度を数え上げた後、あらかじめ決めた種類の数ずつ異なり biterm を分割し、この分割 (セグメント) ごとに N_k と $N_{w|k}$ の更新を行う。本研究では、あらかじめ決めたミニバッチに含まれる異なり biterm の種類数をセグメントサイズとし、提案手法 2 を“ミニバッチ-セグメント学習”と呼ぶ。ここで、ミニバッチ-セグメント学習におけるセグメントは、ミニバッチ学習のミニバッチと実質的に対応している。図 4.3 に示すように、提案手法 1 では、サンプリングした biterm 全てを用いて N_k と $N_{w|k}$ を更新することに対し、提案手法 2 ではサンプリングしたユニークな biterm を複数のセグメントに分割し、このセグメントごとに N_k と $N_{w|k}$ の更新を行う。これにより、提案手法 1 の場合と比較して、サンプリングした biterm 数が同一の状態でもより細かい頻度で N_k と $N_{w|k}$ の更新を行うことができるため、トピック学習の進行がより高速になると考えられる。

4.2.2 単位時間あたりのトピック出現量の計算と近似

本研究では、各単位時間のトピック出現量を求めるために、その時間に投稿されたツイートに含まれる biterm に対して BTM で抽出したトピックの割合を計算し、ツイートに含まれるトピック k の割合の単位時間あたりの合計を式 (4.14) によって計算する。

$$N_{\mathbf{D}_{t,k}} = \sum_{b_v \in \mathbf{B}_{\text{unique}_t}} P(z|b_v) \omega_{t,b_v}. \quad (4.14)$$

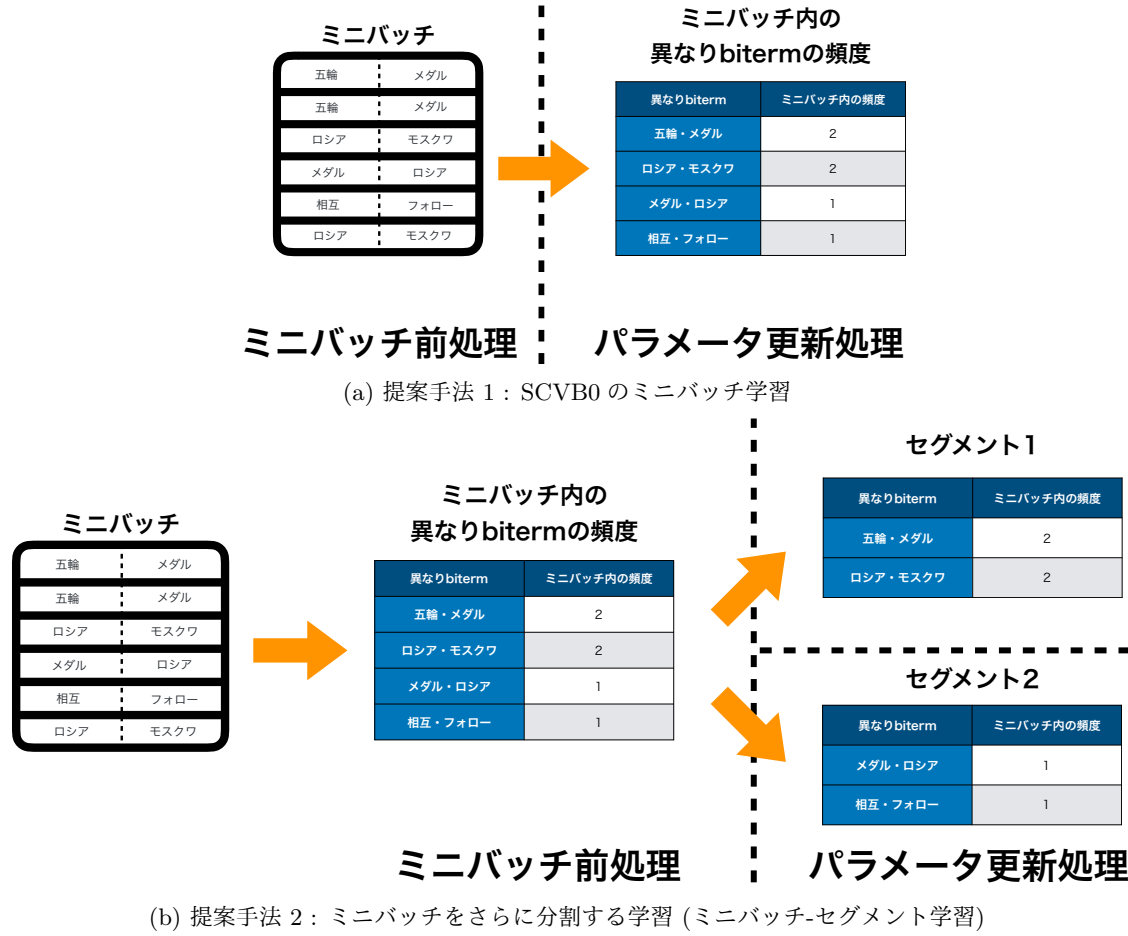


図 4.3: SCVB0 におけるミニバッチ学習とミニバッチ-セグメント学習。提案手法 1 ではミニバッチ全体でパラメータ更新処理を行うことに対して，提案手法 2 ではミニバッチをさらにセグメントに分割してセグメントごとにパラメータ更新を行う。

$\mathbf{B}_{\text{unique}_t}$ は，時刻 t で発生するユニークな biterm の集合を表す．この計算において， ω_{t,b_v} はツイートに含まれる biterm b_v の割合を時刻 t において合計した値であり，本研究では単位時間あたりの“biterm 重み”と呼ぶ．単位時間あたりの biterm 重みは式 (4.15) によって計算される．

$$\omega_{t,b_v} = \sum_{d \in \mathbf{D}_t} P(b_v|d). \quad (4.15)$$

一方で式 (4.14) の計算には，各単位時間においてその時間に投稿されたツイートに含まれる全ての biterm のトピック割合を計算する必要がある．この際，単位時間に出現した全てのツイートに含まれる biterm に対して，トピック割合を計算するために多くの処理時間を有すると考えられる．

各単位時間のトピック出現量の計算時間を短縮するために，本研究では，各単位時間においてその時間に投稿されたツイートに含まれる biterm から一部を取り出し，取り出された biterm のトピックから，各トピックの出現量の計算を行う．特に，特定の時間に投稿されたツイートに含まれる biterm の一部を取り出す方法として以下の手法から比較・検討を行う．

- biterm 重みの離散分布からサンプリングする.
- ユニークな biterm からランダムに選択する.
- biterm 重みの高い順に選択する.

biterm 重みの離散分布からサンプリングする手法では, 単位時間あたりの biterm 重みを異なり biterm ごとの離散分布とみなし, この分布からあらかじめ決めた回数だけランダムにサンプリングを行い, このサンプリングされた biterm の集合から異なり biterm ごとの頻度を集計したものを, 近似した biterm 重みとしてトピック出現量を計算する. biterm 重みは, ある biterm に関する単位時間あたりの平均ツイート数と表しているため, この離散分布から抽出されたサンプルは 1 ツイートとみなすことができる. ユニークな biterm からランダムに選択する手法では, 異なり biterm をランダムに一定数選択し, トピック出現量の計算では元の biterm 重みを使う. biterm 重みの高い順に選択する手法では, 異なり biterm 数が一定になるまで, biterm の高い順に異なり biterm を取り出す. ユニークな biterm からランダムに選択する手法と同様に, トピック出現量の計算では元の biterm 重みを使う. 以上の手法は, いずれも全ての異なり biterm によって計算されたトピック出現量に対する近似したトピック出現量を計算する手法である.

4.3 評価実験

提案手法の有効性を検証するため, 実際のツイートデータから提案手法によってトピック出現量の推定を行う.

4.3.1 実験データ

実験では, 2012 年 8 月 1 日から 2012 年 8 月 7 日の間に投稿されたツイートデータを用いる. これらのツイートデータの総数は 286,223,678, 1 日あたりの平均ツイート数は 40,889,096 である. 本研究では, ツイートに含まれるストップワードを除去するために, ツイートデータに含まれる語彙に対して以下の制約を設ける.

1. 語彙の品詞は, 固有名詞・一般名詞・サ変接続の名詞のみ考慮する.
2. 語彙は, 2 字以上の漢字・ひらがな・カタカナ・数字の組みあわせで構成されている.
3. リツイートを示す「RT」・リプライを示す「@ユーザ名」・ハッシュタグを示す「#」に続く文字列および URL を除く.
4. 「w」や「笑」といった笑いを意味する俗語を除く.

この結果, 実験に使用するツイートデータの総数は 243,600,235 で, 1 日あたりの平均ツイート数は 34,800,034 である. 上記の制約を満たしたツイートデータの単語数について集計した結果を図 4.4 に示す. 図 4.4 より, 含まれる単語の数が 10 以下であるツイートが多く割合を占めることがわかる.

BTM の学習のためにこれらのツイートから biterm の抽出を行う. 抽出された biterm の総数は 3,077,224,154 であり, 異なり biterm 数は 355,980,502 である. 本研究では, データ数削減のために抽出された biterm の頻度の下限を 10 とする. この結果, 実験に用いる biterm の総数は 2,453,945,076, 異なり biterm 数では 32,016,826 である.

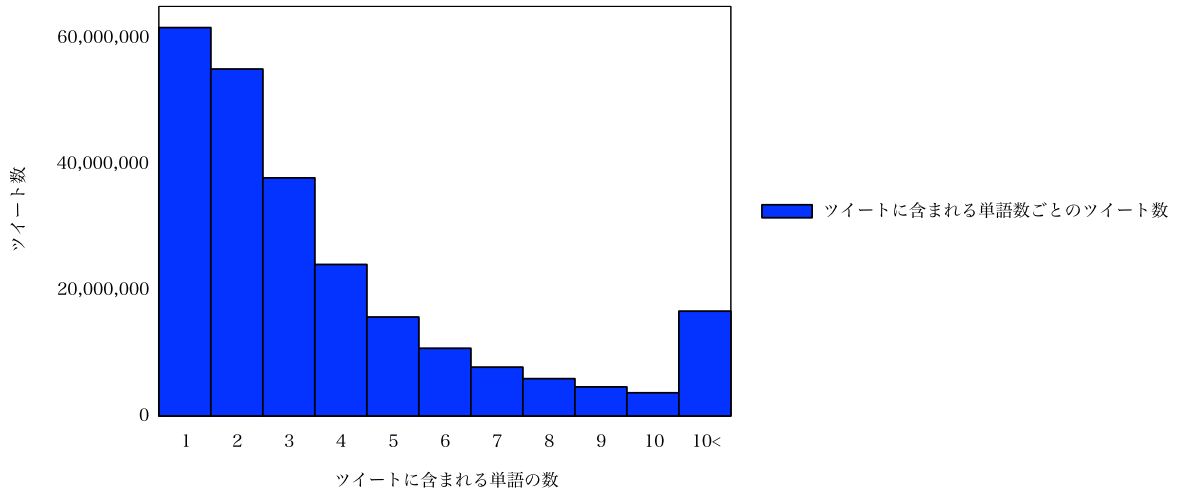


図 4.4: ツイートに含まれる単語数ごとのツイート数

実験環境は、OS が Ubuntu 16.04, CPU が Intel Xeon E5-2630 (2.40GHz) 8core 2 機である。前処理における形態素解析器は MeCab [28], 各アルゴリズムの実装は Python で行った。

4.3.2 トピック学習の高速化の評価

提案手法によってトピックの学習が高速化されていることを検証するため、既存手法との比較を行なった。

比較手法

実験データとして抽出した biterm に対して、以下の 3 種類の手法によってトピックを学習する。

SCVB0 提案手法のベースとなる BTM における SCVB0 の推論アルゴリズムを、ベースラインとして用いる。この手法では、biterm を一つランダムに抽出し、式 (4.4) および (4.5) によって N_k と $N_{w|k}$ の更新を行う。

提案手法 (ミニバッチ学習) 与えられた biterm からあらかじめ設定したミニバッチサイズのみだけ biterm を抽出し、式 (4.12) および (4.13) によって N_k と $N_{w|k}$ の更新を行う。この時、ミニバッチ内の biterm の頻度をあらかじめ数え上げ、各 biterm のトピックの推論は、異なり biterm のごとに行い、推論したトピックと biterm の頻度との積を計算して、その値を N_k と $N_{w|k}$ に更新する。ミニバッチサイズが 1 の時、比較手法となる SCVB0 と同一の手法となる。

提案手法 (ミニバッチ-セグメント学習) 与えられた biterm からミニバッチサイズのみだけ biterm を抽出し、ミニバッチ内の biterm の頻度をあらかじめ数え上げる。そして、あらかじめ決めた異なり biterm をセグメントサイズずつ分割し、セグメントごとにその要素である biterm のトピックを推論し、推論したトピックと biterm の頻度との積を計算して、その値を N_k と $N_{w|k}$ に更新する。セグメントサイズが 1 の時、ミニバッチ学習と同一の手法となる。

パラメータ設定

提案手法および SCVB0 におけるパラメータの設定について説明する．本研究では，BTM のトピック数 K を提案手法および SCVB0 で 512 で固定する．また BTM においてハイパーパラメータとなる α と β はそれぞれ $\alpha = 50/K$ と $\beta = 1/W$ とする．ただし， α は BTM の学習と同時に式 (4.11) によってトピックごとに更新される．SCVB0 および提案手法における減衰重み ν に関するパラメータである τ と κ は，Awaya らの実験と同一のパラメータである $\tau = 1,000$ と $\kappa = 0.8$ とする．

実験方法

本研究では提案手法が，SCVB0 と比較して，モデルの汎化性能が劣らず，処理時間が短縮されているか検証する．検証では，各手法の学習時に抽出される総 biterm 数が同一な条件において，実験データから復元抽出によって biterm を抽出し学習用データと評価用データに分割し，学習用データで学習したモデルに対する評価用データの平均対数尤度と学習に掛かる処理時間を比較する．本研究では，評価用データに対する平均対数尤度を式 (4.16) によって計算する．

$$\text{平均対数尤度} = \frac{1}{|\mathbf{B}_{\text{test}}|} \sum_{b_i \in \mathbf{B}_{\text{test}}} \log \sum_{k=1}^K \theta_k \phi_{k,w_{i,1}} \phi_{k,w_{i,2}}. \quad (4.16)$$

本研究では，実験用に 1 億個の biterm をサンプリングし，学習用データおよび評価用データとして 9 : 1 の割合で分割する．各手法で学習時にサンプリングされる総 biterm 数は 1 億とし，トピック分布のハイパーパラメータの最適化は biterm を 1,000 万個サンプリングするごとに行う．ハイパーパラメータの最適化は一度に 20 回の反復計算を行う．SCVB0 では，1 サンプルの学習を 1 イテレーションとして扱い，このイテレーションを 1 億回行う．提案手法のミニバッチ学習では，ミニバッチサイズ分の学習を 1 イテレーションとして扱い，ミニバッチサイズとイテレーション回数の積が 1 億になる組み合わせで行なう．提案手法のミニバッチ-セグメント学習では，ミニバッチサイズを 100 万，イテレーション回数を 100 に固定し，セグメントサイズを変更する．各手法では，alias 法 [32] [27] によってミニバッチとして biterm をサンプリングする．alias 法における離散分布は，各 biterm の出現頻度をその総和から割った値によるカテゴリカルな分布とする．また，各手法では， N_k と $N_{w|k}$ の更新で使われる式 (4.1) による biterm のトピック確率の計算を，ループ処理ではなく，行列やベクトルに変形し，更新で使われる全ての biterm に対して一度に計算する．これにより，処理速度がプログラミング言語に依存するループ処理ではなく，効率的な数学演算ライブラリで処理されるため，トピック学習の処理時間の高速化が期待される．本研究では効率的な数学演算ライブラリとして，Intel Math Kernel Library¹を用いる．

実験結果

各手法における学習用データを学習させた時の評価用データの平均対数尤度と学習に掛かる処理時間を表 4.1 に示す．ベースラインである SCVB0 と比較して，提案手法であるミニバッチ学習およびミニバッチ-セグメント学習の処理時間はいずれの条件においても短縮されていることがわかる．一方で平均対数尤度は，パラメータによって優劣に差が生じて

¹<https://software.intel.com/en-us/mkl>

表 4.1: 各手法における学習用データを学習させた時の評価用データの平均対数尤度と学習に掛かる処理時間

手法	ミニバッチサイズ	イテレーション	セグメントサイズ	平均対数尤度	処理時間 [sec]
SCVB0	1	100,000,000		-19.231	10496 ± 103
ミニバッチ学習	10	10,000,000	-	-19.111	6123 ± 38
	100	1,000,000		-19.059	6128 ± 22
	1,000	100,000		-19.046	6610 ± 53
	10,000	10,000		-19.088	7253 ± 154
	100,000	1,000		-19.706	7077 ± 174
	1,000,000	100		-21.517	6315 ± 149
ミニバッチ-セグメント学習	1,000,000	100	10	-18.998	5797 ± 144
			100	-18.987	4822 ± 9
			1,000	-18.995	5282 ± 123
			10,000	-19.100	5931 ± 143
			100,000	-19.875	6074 ± 40

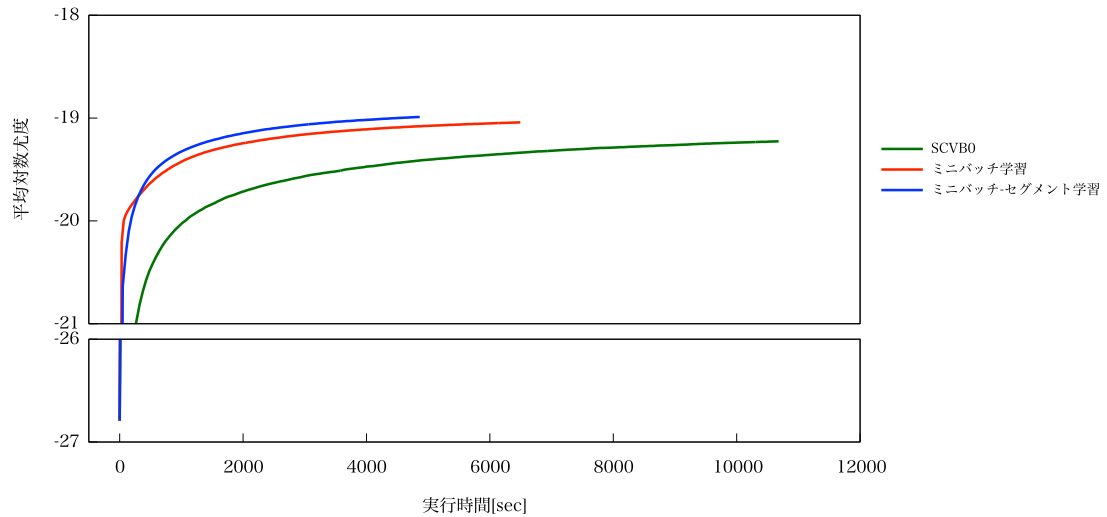


図 4.5: 各手法の実行時間と平均対数尤度の変化

いる。それぞれの手法で最も平均対数尤度の高いパラメータ条件は、ミニバッチ学習ではミニバッチサイズ 1,000・イテレーション 100,000、ミニバッチ-セグメント学習ではセグメントサイズ 100 の場合であり、共に SCVB0 よりも平均対数尤度が高い。特にミニバッチ-セグメント学習でセグメントサイズが 10・100・1,000 の場合、SCVB0 およびミニバッチ学習の全ての条件と比較しても、平均対数尤度が高く、処理時間も短い。このことから、ミニバッチサイズに対して小さなセグメントサイズを設定することによって、汎化性能が向上し、処理時間が短縮されることが示唆される。

SCVB0・ミニバッチ学習 (ミニバッチサイズ 1,000 かつイテレーション 100,000)・ミニバッチ-セグメント学習 (セグメントサイズ 100) における学習の実行時間に対する平均対数尤度の変化を図 4.5 に示す。図 4.5 より、SCVB0 と比較して、ミニバッチ学習およびミニバッチ-セグメント学習は実行時間の経過に対する平均対数尤度の収束が早いことがわかる。またミニバッチ学習とミニバッチ-セグメント学習を比較した場合、学習の初期段階ではミニバッチ学習の方が平均対数尤度が高く、実行時間が 324 秒経過した後ではミニバッチ-セグメント学習の方が平均対数尤度が高くなる。このことから、ミニバッチ-セグメント学習の方が長期的な学習に適していると考えられる。

4.3.3 biterm の一部を抽出しトピック出現量を近似する方法の評価

トピック出現量の近似において、元のトピック出現量に対して最も損失が少なく処理時間が短い手法について比較・検討を行う。

実験方法

本研究で、比較・検討を行う手法は以下の手法である。

- biterm 重みの離散分布からサンプリングする。
- ユニークな biterm からランダムに選択する。
- biterm 重みの高い順に選択する。

検証では、式 (4.14) によって計算された各単位時間におけるトピック出現量に対して、以上の手法で近似したトピック出現量との平均平方二乗誤差 (Root mean square error; RMSE) とトピック出現量の計算に掛かる処理時間を比較する。トピック出現量を計算するトピックは、4.3.2 節の実験において最も平均対数尤度の高いトピックが抽出されたミニバッチ-セグメント学習で推定する。ミニバッチ-セグメント学習では、実験データである 32,016,826 種類の異なり biterm から学習を行い、4.3.2 節の実験と同様に、学習した異なり biterm における平均対数尤度を比較して、最も平均対数尤度の高い学習の設定を確認する。この結果、ミニバッチサイズ 100 万・イテレーション 100 を固定した条件でセグメントサイズを 1,000 に設定した時、最も平均対数尤度が高くなった。したがって本研究では、ミニバッチサイズ 100 万・イテレーション 100・セグメントサイズ 1,000 の設定で行なったミニバッチ-セグメント学習によるトピックからトピック出現量を計算する。

近似したトピック出現量 $\hat{N}_{D_{t,k}}$ は元のトピック出現量 $N_{D_{t,k}}$ に対して単位時間あたりのトピックの出現量の総和が異なるため、式 (4.17) によってスケールを揃えた近似値 $\tilde{N}_{D_{t,k}}$ を求める。

$$\tilde{N}_{D_{t,k}} = \frac{\sum_{k'=1}^K N_{D_{t,k'}}}{\sum_{k'=1}^K \hat{N}_{D_{t,k'}}} \hat{N}_{D_{t,k}}. \quad (4.17)$$

本研究では、全ての biterm によって計算されたトピック出現量 $N_{D_{t,k}}$ に対する $\tilde{N}_{D_{t,k}}$ の RMSE の平均値を式 (4.18) で計算する。式 (4.18) による RMSE の平均値は単位時間あたりの各トピックの出現量の誤差の平均値を意味する。

$$\text{平均 RMSE} = \frac{1}{T} \sum_{t=1}^T \sqrt{\frac{1}{K} \sum_{k=1}^K (N_{D_{t,k}} - \tilde{N}_{D_{t,k}})^2}. \quad (4.18)$$

各手法で単位時間におけるトピック出現量の計算時に使用される異なり biterm 数は 100 万個とする。biterm 重みの離散分布からサンプリングする手法では、異なり biterm 数が 100 万個を超えるまで biterm のサンプリングを続ける。またこの手法では、サンプリングに非常に多くの時間が掛かることが予想されるため、異なり biterm 数が 10 万個と 50 万個に設定した場合も比較する。この手法では、alias 法 [32] [27] によって biterm をサンプリングする。alias 法における離散分布は、各 biterm の biterm 重みをその総和から割った値によるカテゴリカルな分布とする。ユニークな biterm からランダムに選択する手法では、異

表 4.2: 元のトピック出現量に対する各手法の RMSE とトピック出現量計算のための処理時間。RMSE および処理時間は 5 回の実験の平均値とする。

手法	異なり biterm 数	平均 RMSE	処理時間 [sec]
全ての biterm 重みを使う (元のトピック出現量)	4,026,528	-	28,351 ± 108
biterm 重みの離散分布からサンプリングする	100,000	60.565	2,263 ± 162
	500,000	22.509	5,503 ± 21
	1,000,000	13.340	10,263 ± 34
ユニークな biterm からランダムに選択する	1,000,000	213.840	7,788 ± 319
biterm 重みの高い順に選択する	1,000,000	217.605	8,526 ± 51

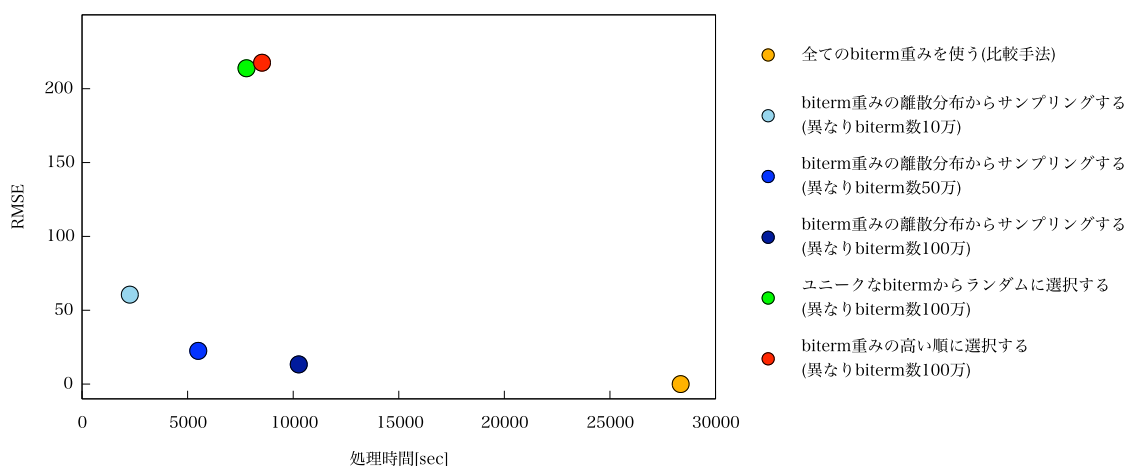


図 4.6: 各手法の処理時間と RMSE の関係。グラフ上の点は表 4.2 の値をプロットしたものである。

なり biterm をランダムに 100 万個選択する。biterm 重みの高い順に選択する手法では、biterm 重みの高い異なり biterm を 100 万個選択する。以上の手法によるトピック出現量の計算をそれぞれ 5 行い、その RMSE と処理時間の平均値によって比較する。ただし biterm 重みの高い順に選択する手法では、一意な RMSE が計算されるため、処理時間のみ平均値を計算する。

実験結果

各手法で計算したトピック出現量における元のトピック出現量に対する RMSE とこの計算に掛かった処理時間を表 4.2 に示す。元のトピック出現量の計算では、異なり biterm 数が単位時間あたりに平均 4,026,528 個存在し、その処理時間は 7 時間 53 分ほど掛かるため、トピック出現量の計算に膨大な時間が掛かっていることがわかる。

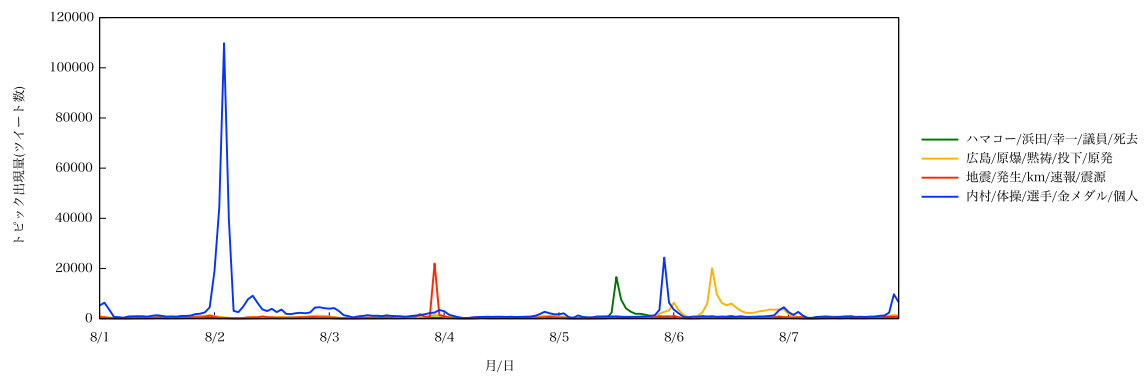
図 4.6 に各手法における RMSE と処理時間の関係を示す。グラフ上の点は表 4.2 の値をプロットしたものである。異なり biterm 数 100 万の条件において、最も処理時間が短い手法は、ユニークな biterm からランダムに選択する手法であった。また同条件において、RMSE が最も低い手法は、biterm 重みの離散分布からサンプリングする手法 (異なり biterm 数 100 万) であった。一方で biterm 重みの離散分布からサンプリングする手法において、用いる異なり biterm 数を少なくした場合、処理時間が減少することが実験結果からわかった。特に、biterm 重みの離散分布からサンプリングする手法 (異なり biterm 数 10

万) は, ユニークな biterm からランダムに選択する手法よりも処理時間が短くなるが, RMSE も低くなることから, より優れた手法といえる. 以上の結果から, biterm 重みの離散分布からサンプリングする手法は他の手法と比べて, 元のトピック出現量に対する誤差が少なく, 高速化できていることがわかる.

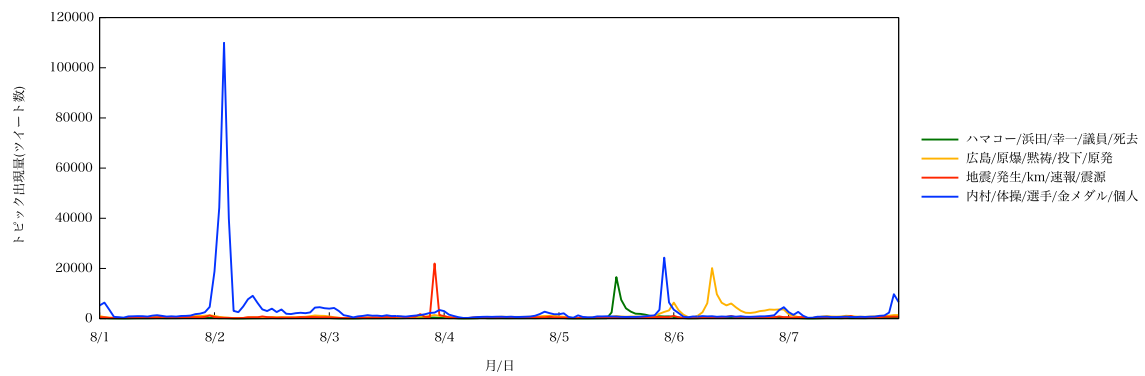
実際に各手法によって計算されたトピック出現量を図 4.7 に示す. いずれの手法もスケールに誤差が生じているが, 時間経過によるトピック出現量の増減に対する誤差が少ないことがわかる. また図 4.7b ではスケールに対する誤差も小さいことがわかる.

4.4 結論

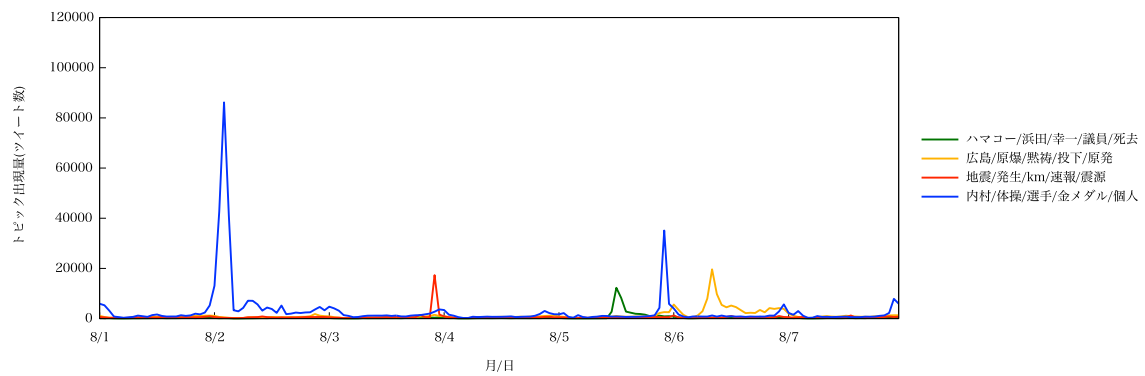
提案手法では, SCVB0 に対して, ミニバッチ学習をベースとした拡張を行うことによって, より高速な学習を行う手法と単位時間内に発生した biterm の中から一部の biterm を使った推論によって, 元のトピック出現量に対する近似を行う手法を提案している. 実際に提案手法を用いて, 2012 年 8 月 1 日から 8 月 7 日に投稿されたツイートからトピックの抽出とそれらのトピック出現量の近似を行なった. この結果, 提案手法によるトピック抽出では既存手法と比較して汎化性能が向上しつつ, 高速な学習が行われていることが確認された. またトピック出現量の近似の比較・検討の結果, 最も損失が少ない近似手法が biterm 重みの離散分布からサンプリングする手法であることが確認され, また抽出する異なり biterm 数を少なくした場合でも, それ以外の手法に対して損失が少ないことがわかった.



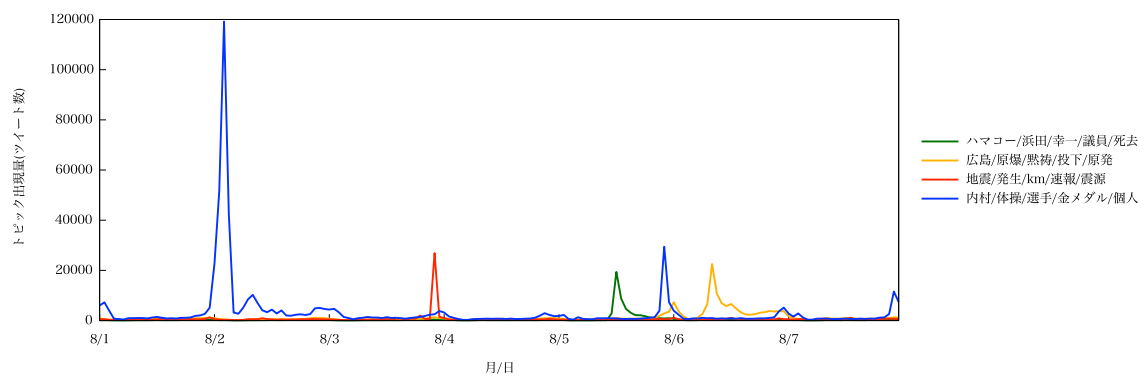
(a) 全ての biterm 重みを使う手法 (元のトピック出現量)



(b) biterm 重みの離散分布からサンプリングする (異なり biterm 数 : 100,000)



(c) ユニークな biterm からランダムに選択する



(d) biterm 重みの高い順に選択する

図 4.7: 各手法によって計算されるトピック出現量. 凡例は各トピックで出現確率の高い語を順に左から並べたものである.

第5章 おわりに

本研究ではツイートデータを対象にトピックを学習し、各トピック出現量の時系列的な変化を抽出する手法を提案した。提案手法では、トピックの学習において Tweet Pooling による擬似文書を学習した LDA と Biterm topic model をそれぞれ利用している。実験では、Tweet Pooling を利用した提案手法によってツイートデータから推定したトピック出現量が現実世界の事象に対するユーザの興味・関心を反映していることを確認した。また BTM を用いる手法では、トピックの学習およびトピック出現量の計算を高速に行う方法を提案し、既存手法と比較して、汎化性能が向上しつつ処理が高速化されていることを確認した。また、トピック出現量を近似する手法として、biterm 重みを離散分布としてサンプリングする手法が、誤差が少なく、処理時間が最も短縮されていることを確認した。

今後の課題は、Tweet Pooling を利用する手法の有効性を一部のトピックの主観的評価のみで行っている点がある。そのため、全てのトピックに対して、現実世界の事象と対応しているか確認を行う必要がある。またこの手法のベースラインとなる推論アルゴリズムを Dynamic topic model に変更した場合のトピックの語彙の一貫性および処理時間の比較を行う必要がある。

今後の展望として、BTM によるトピックの継続的な利用への対応が考えられる。ツイートのトピックを継続的に追跡する場合、提案手法では何度もモデルを学習し直す必要があるため、時間的なコストが掛かる。このことから、一度使用したモデルを再利用する手法を考案し、継続的な利用が可能であるか検証を行う必要があると考えられる。

謝辞

本研究を進めるにあたり、ご指導・ご鞭撻くださいました図書館情報メディア系若林啓助教授に心より感謝いたします。日頃のディスカッションの中で、時には熱い議論になった時も、的確なご指摘を頂いたおかげで本論文を執筆することができました。大学院からの研究室所属で、研究活動としてはスロースタートとなってしまいましたが、この2年間で得られた経験は私自身の人生にとって掛け替えのないものとなっています。

筑波大学図書館情報メディア系佐藤哲司教授には、学内の研究発表会において、研究内容に関する多くの的確なご指摘を頂きました。ありがとうございます。筑波大学図書館情報メディア系手塚太郎准教授には、日頃の合同ゼミにおいて、研究内容に関するアドバイスを頂き、心より感謝いたします。また今年は、手塚・若林研究室として、再スタートする節目の年であり、研究室一丸となって研究活動を行うことができました。研究室のメンバーとの議論はとても刺激的であり、私一人では獲得できない多様な価値観を学ぶことができました。本当にありがとうございました。

参考文献

- [1] Harry Wallop. Japan earthquake:how twitter and facebook helped, 3 2011.
- [2] 芥子育雄, 鈴木優, 吉野幸一郎, 大原一人, 向井理朗, 中村哲. 単語・パラグラフの分散表現を用いた twitter からの日本語評判情報抽出. 第 8 回データ工学と情報マネジメントに関するフォーラム, 2016.
- [3] 荒牧英治, 増川佐知子, 森田瑞樹. 事実性判定を用いたインフルエンザ流行予測. 研究報告音声言語情報処理, 第 2011-SLP-86 巻, pp. 1–8, 2011.
- [4] Jon Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, Vol. 7, No. 4, pp. 373 – 397, 2003.
- [5] 丹羽智史, 土肥拓生, 本位田真一. Folksonomy の 3 部グラフ構造を利用したタグクラスタリング. In *SIG-SWO-A602*, pp. 0701 – 0708, 2006.
- [6] Michael I. Jordan David M. Blei, Andrew Y. Ng. Latent dirichlet allocation. *Journal of Machine Learning Research*, Vol. 3, pp. 993–1022, 2003.
- [7] David Blei and John Lafferty. Dynamic topic models. In *ICML '06 Proceedings of the 23rd international conference on Machine learning*, pp. 113–120, 2006.
- [8] Daichi Koike, Yusuke Takahashi, Takahito Utsuro, Masaharu Yoshioka, and Noriko Kando. Time series topic modeling and bursty topic detection of correlated news and twitter. In *International Joint Conference on Natural Language Processing*, pp. 14–18, 2013.
- [9] Jian Tang, Zhaoshi Meng, Xuanlong Nguyen, Qiaozhu Mei, and Ming Zhang. Understanding the limiting factors of topic modeling via posterior contraction analysis. In *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32 of *Proceedings of Machine Learning Research*, pp. 190–198, Beijing, China, 2014. PMLR.
- [10] David Mimno, Matt Hohnman, and David Blei. Sparse stochastic inference for latent dirichlet allocation. In *the 29th International Conference on Machine Learning*, 2012.
- [11] N. Awaya, J. Kitazono, T. Omori, and S. Ozawa. Stochastic collapsed variational bayesian inference for biterm topic model. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 3364–3370, 2016.
- [12] Tomoharu Iwata, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. Online multiscale dynamic topic models. In *KDD '10 Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 663–672, 2010.

- [13] Tomoharu Iwata, Shinji Watanabe, Takeshi Yamada, and Naonori Ueda. Topic tracking model for analyzing consumer purchase behavior. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI'09*, pp. 1427–1432, San Francisco, CA, USA, 2009.
- [14] Wray Buntine Lexing Xie Rishabh Mehrotra, Scott Sanner. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *SIGIR'13*, 2013.
- [15] Oren Tsur, Adi Littman, and Ari Rappoport. Efficient clustering of short messages into general domains. In *International Conference on Weblogs and Social Media (ICWSM)*, 2013.
- [16] 井上優作, 若林啓. 表記の多様性を考慮したハッシュタグ推薦. 第14回日本データベース学会年次大会, 2016.
- [17] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. BTM: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 12, pp. 2928–2941, 2014.
- [18] 水沼友宏. Twitterにおけるバーストの生起要因と類型化に関する分析. Master's thesis, 筑波大学, 2014.
- [19] Yuji Nakai Shimizu Tohru Kentaro Shimizu Koji Kadota, Ye Jiazhen. Roku: An improved method for the detection of tissue-specific expression patterns. *BMC Bioinformatics*, 2006.
- [20] 石川栄介. 棄却検定の比較表. 岩手大学学芸学部研究年報, Vol. 15, No. 2, pp. 1–7, 1960.
- [21] Smeeton Nigel A Sprent Peter. Applied nonparametric statistical methods. *Chapman and Hall*, p. 480, 1993.
- [22] Sprent Peter. Data driven statistical methods. *Chapman and Hall*, p. 406, 1997.
- [23] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. A Probabilistic Model for Bursty Topic Discovery in Microblogs. *Twenty-Ninth AAAI Conference on Artificial Intelligence*, No. 6, pp. 353–359, 2015.
- [24] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, Vol. 101, No. Suppl. 1, pp. 5228–5235, 2004.
- [25] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic Variational Inference. *JMLR*, pp. 1303–1347, 2013.
- [26] Kento Nozawa and Kei Wakabayashi. Scalable algorithm for probabilistic overlapping community detection. In *Proceedings of the 1st Workshop on Scholarly Web Mining, SWM '17*, pp. 9–16, New York, NY, USA, 2017.
- [27] Jingbo Wang, Wai Wan Tsang, and George Marsaglia. Fast generation of discrete random variables. *Journal of Statistical Software*, Vol. 11, , 2004.

- [28] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis,. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pp. 230–237, 2004.
- [29] Yee Whye Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS’06, pp. 1353–1360, Cambridge, MA, USA, 2006.
- [30] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI ’09, pp. 27–34, Arlington, Virginia, United States, 2009.
- [31] Hanna M. Wallach, David M. Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pp. 1973–1981. Curran Associates, Inc., 2009.
- [32] Alastair J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Softw.*, Vol. 3, No. 3, pp. 253–256, 1977.